Defense Logistics Agency's Year 2000 Program
Managing Organization-Wide Conversion and Compliance

Sarah J. Reed
Defense Logistics Agency System Design Center

The Defense Logistics Agency considers the year 2000 (Y2K) problem mission-critical, and we have treated it as such in planning and executing the largest maintenance effort we have undertaken. The agency kicked off a formal Y2K project in November 1995 with nearly a full year of planning, preparation, and piloting. This article discusses our Y2K initiative and our experiences in raising awareness and in assessing, renovating, and validating our systems. Our program, built on available industry research and our experiences, has been modified as we have gained fresh insight and assimilated new lessons learned.

The Defense Logistics Agency (DLA) maintains 40 million lines of code within 125 automated information systems. Over 80 percent of our major systems make use of date data in some way; half of those date references are high impact; that is, likely to adversely affect comparisons, calculations, or sort processes. DLA's System Design Center (DSDC) is a fee-for-service activity, and funded as we are by disparate functional proponents, we have had rare occasion to conduct a maintenance effort of this magnitude *across* multiple system structures. The degree of planning and coordination called for to accomplish Y2K-compliant systems has been for us, unprecedented. We began our formal Y2K project in November 1995.

System Impact Assessment

The purpose of conducting an organization-wide Y2K impact assessment was to ascertain the magnitude of the Y2K problem and to define and prioritize the remediation requirement. The assessment began in January 1996 with the development of a questionnaire from a Y2K project kick-off session and from industry and government resources. After several revisions, we had 27 questions about date practices, levels of customer awareness, staff support and the application environment (available in the Internet version of this article and on a secured server at http://www.dsdc.dla.mil/priv/projects/year2k/year2k.html). We then identified points of contact for system-level interview and survey. Responses were entered into a database, scrubbed for consistency, then queried and analyzed. Data analysis began in mid-April 1996 and ended with publication of the final report in late June 1996. The extensive data collection and analysis effort resulted in a fairly comprehensive application portfolio profile for DLA standard automated information systems (AIS).

Our Application Profile

A summarized look at some of the component data provides an idea of our size:

Component	Total number
Lines of code	39,577,427
Programs	60,060
Screens	33,416
Reports	8,905
Files	236,271
Database tables	10,379

We have 77 languages, the top languages being COBOL, C, and Assembler. Top database systems are Oracle, Rdb, and Unify. We use 35 different hardware platforms, 16 operating systems, and 311 commercial software packages for application development and maintenance.

Findings, Recommendations, and Resulting Actions

The analyzed systems fell into the following risk classifications determined according to the various remediation actions appropriate for them.

Expected to remain in place beyond 2000, but not compliant (47.1 percent, 56 applications). We addressed this critical risk category by developing funding proposals, conversion strategies and plans, and processes and tools to facilitate analysis, renovation, and testing. Conversion progress is assessed on a regular basis through status reports, progressed project plans, and metric analysis. These actions are intended to mitigate the risk of project slippage on Y2K conversions.

Expected to be rehosted, reengineered, or replaced but not compliant (16 percent, 19 applications). If replacement initiatives experience schedule slippage, Y2K failures could occur within the existing systems targeted for replacement. We regularly track replacement schedules so that we can recommend the initiation of renovation or contingency planning should the replacement system initiatives experience schedule slippage. Contingency planning, an omission perhaps in the early phases of our project, has increasingly become a concern. We now acknowledge the need to plan system-level approaches to fix Y2K problems in a system before its scheduled conversion and deployment has occurred and business area proponents began development of these plans in June 1997. The replacement system progress must also be monitored to ensure Y2K compliance before deployment.

Expected to remain in place beyond 2000 and compliant (33.6 percent, 40 applications). Systems thought to be compliant were perceived to be at less risk than those that were not compliant. However, we insist that system compliance statements be supported with the submission of a certification checklist that describes the system and the Y2K-related testing effort.

Related Observations and Recommendations

AISs not included in the survey data. Several systems were dropped from the survey for a variety of reasons, e.g., they were no longer in existence, are now maintained elsewhere or were too early in development to be profiled. These systems were rechecked a year later to ensure no omissions had occurred.

Database Management Systems (DBMS). The 15 different database systems in use have date types and uses that may or may not be Y2K compliant, depending on how they are used. We discovered that having a date field with a four-position year standard in a DBMS does not make an application that is integrated with it automatically compliant. We have investigated and documented our findings relevant to the date-handling procedures for various DBMSs. Where the results of our assessment reveal noncompliant commercial DBMS products we plan to discontinue its use or obtain a version upgrade, as appropriate.

Interface recommendations. Initially, the Y2K Program office maintained that all interfaces should be identified, described and documented, then negotiated to determine format transition actions. A vociferous outcry from the project managers of the largest AISs and advice from our executive Configuration Control Board (CCB) convinced us to concentrate resources on renovation of the system and filter building to protect our systems from noncompliant incoming data—and not on self-initiated interface format changes. Ultimately, we did agree that an interface strategy was necessary even if our base premise would be not to change the interface formats until receipt of a specific requirement. This strategy was delivered in August 1997.

Release Management. AIS upgrades are normally released to multiple deployment sites. Y2K considerations presented the additional challenge of simultaneously implementing Y2K releases, nonyear-2000 releases, and in some cases, mixed releases. To facilitate adequate implementation planning, we worked to establish a liaison with our chief deployment site.

Date formats. The wide variety of date formats (48) used among the AIS points out the need for greater standardization of date formats, but conversion to four-position year formats is not always practical for large, complex legacy systems given the risk and time constraints associated with them. Because of the variety of date formats and the prevalence of two-digit years, we recommended common date modules for processing and provided these modules for our more prevalent languages.

Vendor product compliance. The application impact assessment highlighted the need to further investigate the compliance of hardware platforms and operating systems. For our organization, compliance status of these commercial products must be established with the product vendor. The Y2K Program office continues to pursue the issue of vendor product compliance and to publish its findings along with appropriate recommendations.

Awareness. In spite of briefs to senior officials, we felt that perhaps customer and user communities were still largely uninformed even as they began funding the initiative in January 1997. We subsequently began to develop white papers and briefings that discussed aspects of interest to the user community. In addition, we now have regular project forums to facilitate the sharing of progress, findings, and concerns throughout the development organization.

Assessment process. We knew going into the assessment process that developing good survey questions, properly targeting the survey, and effectively analyzing the results would be a difficult task for our team, none of whom had a statistical analysis or survey science background. We were also concerned about the quality of the survey responses, often subjective and estimated. Though satisfied with our results and subsequent recommendations that guided our later efforts, we would recommend that inclusion of a team member with a statistical science background would be of benefit during impact assessment.

Vendor Product Risk Assessment and Mitigation

In April 1996, we laid out the following high-level plan for conducting a vendor product assessment:

- Send letters to all vendors for whom we had products under maintenance and ask for a statement regarding Y2K compliance status of the products we held.
- Enter response data into a repository.
- Analyze for risk classification.
- Recommend appropriate mitigation actions.
- Track and report outcomes and status.

Although almost 90 percent of the vendors responded, we began to discover a need to collect much more information. We had collected some vendor product information during the impact assessment; however, not all respondents provided clear or complete information about the commercial products integrated with their applications. This prompted a call for additional information.

Identification of Development Environment Software

In November 1996, we provided our first report on Y2K compliance in our vendor products. The resulting recommendations were several, but the chief recommendation was to expand upon our original data collection effort. We still could not confidently describe the risk in our hardware and system software, because we did not have an adequate baseline picture. Where we had expected to find a similar identification of products between our organizations, we found that of 74 products identified on our chief mainframe machine, 53 were *not* previously identified by the application support areas or the acquisition office. We established possible reasons for the discrepancies, then arduously worked to resolve them.

Where Are We Now?

There were 276 products identified as lacking complete compliance information. If the lack of information was because we could not properly identify the vendor, the AIS support group was contacted to help identify the product vendor to obtain the required information. Vendor contact was established or re-established to obtain new or additional information product information.

Improve software portfolio management processes. The great discrepancies between the view of what is owned, installed, and used was of such concern to us that we launched a major software portfolio management improvement effort, wider in scope and more permanent in its legacy than the Y2K effort.

Follow up with vendors expected to provide compliant products at a future date. For these products, we notify groups who support applications integrated with these products, because the projected date of compliance may be unacceptably late. We also make them aware of product upgrade impacts upon their application. Upgrade strategies will be developed on a case-by-case basis to manage application impact and ensure timely arrival, installation, and testing of the upgraded product.

Develop independent strategy for IBM products. A planned operating system upgrade should replace several, but not all, currently held noncompliant products. It was determined that the Year 2000 Program and OS/390 upgrade project team would need to work closely together to accomplish the OS upgrade and Y2K integration testing in a timely manner.

Develop testing strategies for those products vendors have stated are compliant. Ascertaining the state of vendor products as we were from direct vendor replies to our questions, research of vendor-supplied World Wide Web statements regarding Y2K compliance, and in some cases, relying on third-party-published studies still leaves a vulnerability regarding the actual performance of the vendor product. We have deter-

Table 1. *Y2K support tool categories according to DLA internal priorities.*

Priority	Tool Capability	Tool Category	Prioritization Rationale
1	Data Generation	Validation	Preparation of test data for year 2000; expected to be a significant effort via the automated generation of test data
	Test Case Management	Validation	Test case management can reduce effort to develop current century, cross-century, and next century test cases.
	Record/Playback Capability	Validation	Recording inputs for future playback reduces re-testing efforts.
	Data Aging	Validation	Preparation of test data for Y2K testing, via the ability to advance dates in test data transactions.
	Clock Simulators	Validation	Ability to simulate dates other than current system date.
2	Bridging (file conversion)	Renovation	Use of tools to reduce the bridging effort.
3	Test Coverage Analysis	Validation	Quantifies program logic execution during test.
4	Field Expansion	Renovation	Lower priority; our strategy utilizes expansion on limited basis due to associated cost and risk.
5	Code Generator	Renovation	Lower priority because capability already exists .
6	Version Control	Config. Mgt.	Capability available.
	Change Control	Config. Mgt.	Capability available.
7	Impact Analyzers	Assessment	Capability available.
	Cost of Work Estimating	Assessment	Capability available.
	Date Subroutines	Renovation	Capability available.
	Automated Conversions	Renovation	Capability available.
8	Version Merging	Renovation	Capability available.
	Change Tracking	Config. Mgt.	Capability available.
	Date Finders	Assessment	Capability available.
	Cross-References	Assessment	Capability available.
	File Comparisons	Validation	Capability available.
9	Data Name Rationalization	Assessment	Capability available.

mined that at least minimal spot testing of in-house versions of products stated to be compliant is prudent.

Review and Selection of Support Tools

The objective of this task was to recommend a tool set for use in performing Y2K assessment, renovation, and validation. Several sources were used to identify potential tools and determine appropriate tools for further investigation —product literature, Web sites, expos and conferences, industry-developed tool reports, and demonstrations. The following requirements were identified:

- Where possible, the tools should be from a major vendor to address concerns about tool maturity and vendor support.
- The tools should primarily address the major languages and platforms used within the DSDC. These were determined to be primarily COBOL running on the MVS platform, with C a distant second.
- Use of the tools should not require an entirely new method; they should require minimal training for effective use, and execute in the environment to which the development staff is accustomed.

Support tool categories are described in Table 1. The higher priority tools (1, 2, 3, etc.) are those tools for which little or no existing internal capability existed at the time of the assessment.

Tool Recommendations and Implementations

Most identified tool needs could be met with tools already owned. Release management changes were recommended to ensure all development sites had the latest tool versions. New tool recommendations were prioritized, and tools were brought in for environmental testing and evaluation prior to purchase. A Y2K analysis and remediation workshop was developed primarily for COBOL programmers, with emphasis on the procedural logic or date windowing solution.

Lessons Learned

- Outdated versions of system software in the development environment limited our ability to fully evaluate Y2K tool offerings.
- Lack of a clear description of the current or targeted development environment made planning for the incorporation of targeted Y2K tools into a long-term engineering strategy difficult.
- Once a product was obtained, it was sometimes difficult to get it installed. We were unable to complete the testing of software during the contract period of performance, and it was frustrating to know how helpful these tools could be to the development staff in accomplishing the Y2K work.

Certification Process to Benchmark and Report System Compliance

Organizational confidence in Y2K compliance statements requires a common certification process and benchmark. As with the impact assessment, a process was initially brainstormed at our first team meeting, then evolved. The draft guidance and certification checklist was piloted on systems believed to be compliant. The pilot results and emerging Department of Defence (DoD)-level guidance caused us to revise the process and checklist before submission to the executive CCB (available in the Internet version of this article and on a secured server at http://www.dsdc.dla.mil/priv/

Table 2. *Characteristics of six DLA Y2K conversion projects.*

Name	Platform Characteristics	Tools Used	Strategy Used	Pilot Size (Lines of Code)	Cost per Line of Code
EMACS	Distributed; 486 UNIX PCs. Unify DBMS, ACCELL, C code, shell scripts, UNIFY report writer, RPT.	UNIFY Perl Script	Expansion	63,000	$.43
DISMS	COBOL, Supra/MANTIS DBMS.	SLAP TSO/ISPF, MANTIS full screen editor, FileAid, ISPF	Procedural	9,177 LOC 20 COBOL, 8 Mantis programs	$5.05
MOCAS Production Redesign	COBOL, MANTIS	SLAP Mandate2 MandateX	Procedural	124,313 LOC, 61on-line, 51 batch programs	$.63
SAMMS Financial Funds History	COBOL	SLAP Mandate	Procedural and Expansion	104,061	$1.06
Technology - Automated Password Change Facility	TSO/ISPF workbench for MVS editing. VI editor for UNIX shell, C modifications. In-house-developed C code scanner	SLAP, MANDATE X, MANDATE C	Procedural	7,843	$2.28
DFAMS AIS-Wide Analysis	COBOL, ALC, a proprietary API, Communications software. Data-Unify, Oracle, and CA-Datacom databases.	Specially developed search tool for the M204 code; SLAP for COBOL	N/A	2,016,672	NA

projects/year2k/year2k.html). The process was based in part on a premise that different situations could call for different certification levels reflecting various degrees of risk. However, after presentation to and feedback from the CCB, we revised the certification documents again, and the idea of multiple certification levels was ultimately dropped.

Certification Process

The process covers internally developed configuration items only. A separate initiative addresses the compliance of vendor-provided software and hardware products. Our deployment environment is under the control of another DoD agency; therefore, our participation in the compliance of the deployment environment is limited to the communication of issues.

The Y2K Program office provides certification guidance and a checklist. The system support group returns the completed checklist and test documentation. The program office verifies incomplete or inconsistent information and recommends a certification action prior to presentation to the customer for final certification.

Lessons Learned

- In spite of our efforts to streamline it, the checklist is lengthy, and although the majority of the questions are "yes or no," determining the answers, unless the process is to be shortchanged, requires a fair amount of testing. System proponents that believe their systems are already compliant tend not to plan resources for Y2K testing and certification. Convincing system support groups and funding proponents of the necessity to certify Y2K compliance has been, and continues to be, difficult.
- Industry guidelines or cost models existed for multiphase conversion efforts but did not for certification-only efforts; thus, further hampering attempts to encourage Y2K effort and cost planning for systems already maintained to be compliant.
- In spite of our best efforts to baseline and keep the certification process stable throughout the duration of the program, it is likely going to change as we evolve the process and as higher levels within DoD become increasingly involved in providing Y2K guidance. This compelling interest from higher levels of our organization has at times caused us to question our early formed tenet that moving ahead without such guidance was preferable to waiting for guidance that might come too late to be of benefit.
- The process was designed to provide Y2K test information without being onerous to the projects. We encouraged the use of established software engineering testing and certification processes to the greatest extent possible. However, in working this cross-organizational initiative, we have found test process inadequacies and inconsistencies across the organization. A common benchmark, such as an organization-wide certification process, is difficult to achieve without the existence of common processes.

Pilot Project Study

The Y2K Program proposed pilot projects as a vehicle to discover early Y2K lessons learned relevant to the Y2K change effort. Three purposes would be served:

- Generation of actual conversion cost metrics.
- Identification of not-yet-anticipated Y2K challenges.
- Jump-start of the conversion effort through these early pilots.

Pilot Project Profile

Table 2 summarizes the characteristics of six Y2K conversion pilot projects.

Lessons Learned

Estimation of the Effort Involved. Of immediate note is the variance in pilot project lines of code (LOC) costs. For smaller (less than 10,000 LOC) samples, the LOC cost was higher. We also found that the lack of a date simulation tool adversely impacted the testing, causing extra delays and a reduction in the quality of testing.

Renovation. The selected approach for making a system Y2K compliant depends on the specific AIS and its design, interfaces, and implementation environment. Although smaller, more modern systems are able to successfully perform date expansion, larger legacy systems with multiple interfaces would require considerable more time and effort to do so, for both modification and testing. Expanding the date data field enables the meeting of standards but may cause considerable down time during implementation—partially because of the lessened ability to incrementally implement converted segments—and may cause cosmetic display challenges. In some cases, a procedural approach is the only risk viable approach to achieving Y2K compliance in time.

Awareness and Communications. A system can be Y2K-compliant without conversion of the user-interface layer, i.e., report and screen displays, the modification of which could increase remediation costs and schedules unacceptably. In one of the pilots, we found that the number of on-line programs that needed to be changed increased from six to 60 with the remediation of the user-interface layer. It is important for the user to understand the impacts of various system, interface, and user interface renovation choices.

Customers and users also need to know that delays in replacement system implementation increases the Y2K failure risk to the existing system. Additionally, intention to replace a legacy system does not entirely relieve the customer of Y2K-related costs; contract modifications and additional testing and certification activities may still be required to produce Y2K-compliant new systems.

At the time of the pilots, effective Y2K communication within DSDC was limited; therefore, tool and technology capabilities used by one pilot were not effectively used by *all* pilots. To compensate for this lack of communication, the program office planned and conducted a series of sessions for projects to share ideas, concerns, and information.

Tools. In-house tool capability was assessed as being sufficient for mainframe-based application assessment and renovation phases. Tools for other platforms are becoming increasingly available. However, problems related to the lack of a robust test environment and date simulation utilities caused delays during pilot testing efforts. This realization from pilot efforts escalated our tool assessment and procurement initiatives.

Testing. Testing time and effort was underestimated for all pilots. In one pilot, test processes were lacking; for example, system and functional test plans were developed but lacked the specificity required to sufficiently address the Y2K concerns. The addition of tools and training, without effective test processes, would probably not be sufficient to ensure success of the effort. Consequently, we incorporated testing expertise to guide the test plan development and implementation, but the responsibility for the planning and implementation remained at AIS level. Each AIS was encouraged to develop a testing strategy for specific needs related to test environments, test tools, test utilities, and test data. The program office provides support in tool procurement, installation, and training.

Release Management. Close coordination with deployment environments is indicated not only to ensure adequate support for the installation of support tool releases but also for a larger-than-normal number of new application releases into the field. Where file conversions are involved, coordination is particularly important as other non-standard DLA systems use file formats passed by our standard systems.

Observations and Conclusions

Throughout this article, we have discussed our approaches, results, and lessons learned as they pertain to significant aspects of our effort. Here we offer some overall observations and conclusions.

Provide a Sufficient Resource Investment

Resourcing Y2K initiatives is not a one-time commitment because resourcing will continuously be a challenge at all levels of Y2K execution. In the current DoD environment, a key strategy for dealing with resource shortages is contracting. But even our contractors are having difficulty retaining enough people to execute their contracts in a tight, competitive labor market. Another challenge we have had in the government is in keeping the noncontracted, what we call organic personnel commitments stable. We are constantly downsizing—even offering incentives for early retirement—and no resource commitment is a sure bet.

Provide Sufficient Investment in Testing

Every pilot incurred slowdowns and obstacles during the testing phase. As discussed, we discovered that variances in testing procedures within the organization affected attempts to baseline a certification process. Effort expended in defining Y2K-specific testing practices and in building adequate test environments are worth the front-end investment.

Be Prepared for High-Level Visibility

We recognize the benefit of high visibility as the crucial components of sponsorship, and commitment can wax and wane without it. However, visibility can have negative impact. We quickly became swamped with project and progress requests for information from various government agencies. In hindsight, designation of a communications officer would have been invaluable in answering these calls for information. Lack of such a resource affected our ability to stay focused on developing strategy and plans and on implementation to the desired degree.

Summary

Following is a recapitulation of what we have found to be the most critical factors affecting our Y2K remediation effort:

- Sustaining organizational commitment.
- Understanding the organizational baseline at the outset of the initiative.
- Understanding the challenge in applying standard remediation solutions across an organization with disparate processes.
- The adequacy of testing tools *and* processes.
- Focus on the underlying hardware and system software is a major component of the total picture. We did not anticipate the level of work and difficulty involved in this aspect at the outset.
- Obviously, remediation for systems with future date projections should be a first priority. But there must also be prioritization and contingency planning *within* the system effort, because failures will surely occur even as the system is being renovated.
- Carving out enough time for integrated testing of renovated applications with upgraded system software and hardware may turn out to be our biggest challenge. ◆

About the Author

Sarah J. Reed began her DoD career with the DSDC in 1987. In that time, she has been involved in or led a number of projects, such as corporate-wide metrics collection, analysis and implementation, the Organizational Culture Initiative, and fee-for-service implementation.

In 1991, she worked with the DLA Information Technology Policy Board representative to review, suggest, and comment on DoD policy, some of which shapes the DoD mission today. In 1992, she led the effort to establish a baseline Capability Maturity Model assessment for the organization. In December 1995, she accepted the challenge to lead the Y2K Program at DSDC. In December 1997, she became a private consultant on Y2K issues.

DSDC's Year 2000 Program office is now being managed by David Koppy. For further information on DSDC's Year 2000 Program please, contact

Cindy Nickoson
DLA Systems Design Center
Columbus, OH 43213
Voice: 614-692-9296 DSN 850-9296
Fax: 614-692-8393 DSN 850-8393
E-mail: cnickoson@dsdc.dla.mil

If you have questions for the author, please contact

Sarah Reed
CompuWare
Voice: 614-847-8212